CP decomposition of semi-nonnegative semisymmetric tensors based on QR matrix factorization

Lu Wang, Laurent Albera, Amar Kachenoura, Huazhong Shu, Lotfi Senhadji

CP decomposition of semi-nonnegative semi-symmetric tensors based on QR matrix factorization

Lu Wang*†§, Laurent Albera*†§¶, Amar Kachenoura*†§, Hua Zhong Shu‡§ and Lotfi Senhadji*†§
*INSERM, UMR 1099, Rennes, F-35000, France †Université de Rennes 1, LTSI, Rennes, F-35000, France
§Centre de Recherche en Information Biomédicale sino-français (CRIBs), Rennes, France
‡LIST, Southeast University, 2 Sipailou, 210096, Nanjing, China
¶INRIA, Centre Inria Rennes - Bretagne Atlantique, 35042 Rennes, France.
Email: wangLyu1986@hotmail.com

***Abstract*—The problem of Canonical Polyadic (CP) decomposition of semi-nonnegative semi-symmetric three-way arrays is often encountered in Independent Component Analysis (ICA), where the cumulant of a nonnegative mixing process is frequently involved, such as the Magnetic Resonance Spectroscopy (MRS). We propose a new method, called $\mathrm{JD}^{+}_{\mathrm{QR}}$, to solve such a problem. The nonnegativity constraint is imposed by means of a square change of variable. Then the high-dimensional optimization problem is decomposed into several sequential rational subproblems using QR matrix factorization. A numerical experiment on simulated arrays emphasizes its good performance. A BSS application on MRS data confirms the validity and improvement of the proposed method.**

I. Introduction and problem formulation

Canonical Polyadic (CP) decomposition of a multi-way array [1]–[3] plays an important role in Blind Source Separation (BSS), particularly in Independent Component Analysis (ICA) [4]. In this paper, we consider the following semi-nonnegative semi-symmetric CP decomposition problem:

Problem 1. *The semi-nonnegative semi-symmetric CP decomposition of a 3-way array $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{N\times N\times K}$, is the minimal linear combination of rank-1 3-way arrays that yields $\boldsymbol{\mathcal{C}}$ exactly:*

$$\boldsymbol{\mathcal{C}} = \textstyle\sum_{p=1}^{P} \boldsymbol{a}_p \circ \boldsymbol{a}_p \circ \boldsymbol{d}_p \tag{1}$$

subject to $\boldsymbol{A} = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_P] \in \mathbb{R}^{N\times P}$ having nonnegative components, where $\circ$ denotes the outer product. $\boldsymbol{A}$ and $\boldsymbol{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_P] \in \mathbb{R}^{K\times P}$ are called the loading matrices of $\boldsymbol{\mathcal{C}}$. P is then the rank of $\boldsymbol{\mathcal{C}}$.

The decomposition is considered to be essentially unique when the uniqueness is guaranteed up to scaling and permutation indeterminacies. This problem is often encountered in ICA when a nonnegative mixing matrix is considered. For example, in Magnetic Resonance Spectroscopy (MRS), the mixing matrix contains the positive concentrations of the source metabolites. Then the 3-way array built by stacking the matrix slices of a cumulant is both nonnegative and symmetric in two modes. Equation (1) can also be described by using the frontal slices of $\boldsymbol{\mathcal{C}}$: $\boldsymbol{C}^{(k)} = \boldsymbol{\mathcal{C}}_{:,:,k} = \boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{\mathsf{T}}, \forall\, k \in \{1, 2, \cdots, K\}$, where $\boldsymbol{D}^{(k)} \in \mathbb{R}^{P\times P}$ is a diagonal matrix whose diagonal contains the elements of the k-th row of $\boldsymbol{D}$, and $\boldsymbol{C}^{(k)} \in \mathbb{R}^{N\times N}$ is the k-th frontal slice $\boldsymbol{\mathcal{C}}$. In this paper, we focus on computing the square matrix $\boldsymbol{A}$, where $N = P$. In order to compute $\boldsymbol{A}$, we can resort to solve the following nonnegative Joint Diagonalization by Congruence (JDC) problem:

Problem 2. *Given a 3-way array $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{N\times N\times K}$ with K symmetric frontal slices $\boldsymbol{C}^{(k)} \in \mathbb{R}^{N\times N}$, find a matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and K diagonal matrices $\boldsymbol{D}^{(k)} \in \mathbb{R}^{N\times N}$ such that:*

$$\forall\, k \in \{1, 2, \cdots, K\},\ \boldsymbol{C}^{(k)} = \boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{\mathsf{T}} \tag{2}$$

subject to $\boldsymbol{A}$ having nonnegative components.

Many existing CP algorithms handle the symmetry and the nonnegativity separately, such as in [5]–[7]. Several methods consider the combination of both constrains [8], [9], but they aim at solving different problems rather than problem 1. Only a few methods were proposed to solve the nonnegative JDC problem [10], [11]. In this paper, we propose a new algorithm, called $\mathrm{JD}^{+}_{\mathrm{QR}}$, based on minimizing the following indirect least square criterion [6], [12]:

$$J_1(\boldsymbol{A}) = \textstyle\sum_{k=1}^{K} \|\, \mathrm{off}(\boldsymbol{A}^{-1}\boldsymbol{C}^{(k)}\boldsymbol{A}^{-\mathsf{T}})\|_F^2 \tag{3}$$

where $\mathrm{off}(.)$ vanishes the diagonal components of the input matrix, the superscript $^{-\mathsf{T}}$ denotes the inverse of the transposed matrix, and $\|.\|_F$ computes the Frobenius norm. The nonnegativity constraint is imposed by means of a square change of variable. The QR matrix factorization of the Hadamard square root of $\boldsymbol{A}$ decomposes the high-dimensional optimization problem into a sequential rational subproblems. In addition, the rotation matrix and the unit triangular matrix of the QR factorization have unit determinants, therefore the resulting matrix $\boldsymbol{A}$ is nonsingular. A numerical experiment on simulated arrays emphasizes its good performance. A BSS application on MRS data confirms the validity and improvement of the proposed method.

II. The $\mathrm{JD}^{+}_{\mathrm{QR}}$ Method

In order to avoid the inverse of $\boldsymbol{A}$ in cost function (3), let us consider the following assumptions: ***i)*** $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times N}$ is nonsingular; ***ii)*** $\boldsymbol{D} \in \mathbb{R}^{K\times N}$ is nonsingular and does not contain zero entries. Then each frontal slice of $\boldsymbol{\mathcal{C}}$ is nonsingular and its inverse can be expressed as follows:

$$(\boldsymbol{C}^{(k)})^{-1} = \boldsymbol{A}^{-\mathsf{T}}(\boldsymbol{D}^{(k)})^{-1}\boldsymbol{A}^{-1} \tag{4}$$

In practice, only the sufficiently well-conditioned matrix $\boldsymbol{C}^{(k)}$ is chosen when its condition number is below a predefined threshold. We use $\boldsymbol{C}^{(k,-1)}$ to denote $(\boldsymbol{C}^{(k)})^{-1}$ for simplicity. Equation (4) shows that $\boldsymbol{C}^{(k,-1)}$ is jointly diagonalizable by $\boldsymbol{A}$. Then $\boldsymbol{A}$ can be estimated by minimizing the following modified criterion of (3) directly:

$$J_2(\boldsymbol{A}) = \sum_{k=1}^{K} \| \operatorname{off}(\boldsymbol{A}^\mathsf{T}\boldsymbol{C}^{(k,-1)}\boldsymbol{A})\|_F^2 \tag{5}$$

The nonnegativity constraint on $\boldsymbol{A}$ can be imposed by a square change of variable: $\boldsymbol{A}=\boldsymbol{B}\boxdot\boldsymbol{B}=\boldsymbol{B}^{\boxdot 2}$, where $\boldsymbol{B}\in\mathbb{R}^{N\times N}$ and where $\boxdot$ denotes Hadamard product [13], [14]. Then we can find $\boldsymbol{A}\in\mathbb{R}_+^{N\times N}$ by estimating $\boldsymbol{B}\in\mathbb{R}^{N\times N}$, such that $\boldsymbol{A}=\boldsymbol{B}^{\boxdot 2}$, and $\boldsymbol{B}$ is the global minimum of the following cost function:

$$J_2(\boldsymbol{B}) = \sum_{k=1}^{K} \| \operatorname{off}\left((\boldsymbol{B}^{\boxdot 2})^\mathsf{T}\boldsymbol{C}^{(k,-1)}\boldsymbol{B}^{\boxdot 2}\right)\|_F^2 \tag{6}$$

Now let us recall the following definitions and lemmas:

Definition 1. *A unit upper triangular matrix is an upper triangular matrix whose main diagonal entries are* 1.

Definition 2. *An elementary upper triangular matrix* $\boldsymbol{R}^{(i,j)}(r_{i,j})$ *is equal to an identity matrix except the* (i,j)*-th entry, which is equal to* $r_{i,j}$.

Definition 3. *A Givens rotation matrix* $\boldsymbol{Q}^{(i,j)}(\theta_{i,j})$ *is equal to an identity matrix except the* (i,i)*-th,* (j,j)*-th,* (i,j)*-th and* (j,i)*-th entries, which are equal to* $\cos(\theta_{i,j})$, $\cos(\theta_{i,j})$, $-\sin(\theta_{i,j})$ *and* $\sin(\theta_{i,j})$*, respectively.*

Lemma 1. *Any* $(N\times N)$ *unit upper triangular matrix can be factorized as a product of* $N(N-1)/2$ *elementary upper triangular matrices.*

Lemma 2. *Any* $(N\times N)$ *orthonormal matrix can be factorized as a product of, at most,* $N(N-1)/2$ *Givens rotation matrices.*

For any nonsingular matrix $\boldsymbol{B} \in \mathbb{R}^{N\times N}$, the QR matrix factorization decomposes it as $\boldsymbol{B}=\boldsymbol{Q}\boldsymbol{R}\boldsymbol{\Lambda}$, where $\boldsymbol{Q}\in\mathbb{R}^{N\times N}$ is a orthonormal matrix, $\boldsymbol{R}\in\mathbb{R}^{N\times N}$ is a unit upper triangular matrix, and $\boldsymbol{\Lambda}\in\mathbb{R}^{N\times N}$ is a diagonal matrix. Due to the indeterminacies of the CP decomposition, the matrix $\boldsymbol{B}$ solving (6) can be chosen as $\boldsymbol{B}=\boldsymbol{Q}\boldsymbol{R}$ without loss of generality. Moreover, lemma 1 and lemma 2 yield that $\boldsymbol{B}$ can be written as a product of the following matrices:

$$\boldsymbol{B} = \prod_{i=1}^{N}\prod_{j=i+1}^{N}\boldsymbol{Q}^{(i,j)}(\theta_{i,j})\prod_{i=1}^{N}\prod_{j=i+1}^{N}\boldsymbol{R}^{(i,j)}(r_{i,j}) \tag{7}$$

As a consequence, the minimization of (6) with respect to $\boldsymbol{B}$ is converted to the estimation of $N(N-1)$ parameters: $\theta_{i,j}$ and $r_{i,j}$. We propose a Jacobi-like procedure, called $\text{JD}_{\text{QR}}^{+}$, in order to compute these parameters sequentially.

A. Minimization with respect to $\boldsymbol{Q}^{(i,j)}(\theta_{i,j})$

Let $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ denote the current estimate of $\boldsymbol{A}$ and $\boldsymbol{B}$ before estimating $\boldsymbol{Q}^{(i,j)}(\theta_{i,j})$, respectively. Let $\tilde{\boldsymbol{A}}^{(\text{new})}$ and $\tilde{\boldsymbol{B}}^{(\text{new})}$ stand for $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ updated by $\boldsymbol{Q}^{(i,j)}(\theta_{i,j})$, respectively. Furthermore, the update of $\tilde{\boldsymbol{B}}$ is defined as follows:

$$\tilde{\boldsymbol{B}}^{(\text{new})} = \tilde{\boldsymbol{B}}\boldsymbol{Q}^{(i,j)}(\theta_{i,j}) \tag{8}$$

In order to compute $\theta_{i,j}$, the natural way is to minimize criterion (6) with respect to $\theta_{i,j}$ by replacing matrix $\tilde{\boldsymbol{B}}$ by $\tilde{\boldsymbol{B}}^{(\text{new})}$. For the sake of convenience, we denote $J_2(\theta_{i,j})$ instead of $J_2(\tilde{\boldsymbol{B}}^{(\text{new})})$. $J_2(\theta_{i,j})$ can be expressed as follows:

$$J_2(\theta_{i,j})=\sum_{k=1}^{K}\left\|\operatorname{off}\left\{[(\tilde{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}]^\mathsf{T}\boldsymbol{C}^{(k,-1)}[(\tilde{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}]\right\}\right\|_F^2 \tag{9}$$

The Hadamard square of $\tilde{\boldsymbol{B}}^{(\text{new})}$ in (9) can be written as a function of $\theta_{i,j}$ as follows:

$$(\tilde{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}=\tilde{\boldsymbol{B}}^{\boxdot 2}(\boldsymbol{Q}^{(i,j)}(\theta_{i,j}))^{\boxdot 2}+\sin(2\theta)(\tilde{\boldsymbol{b}}_i\boxdot\tilde{\boldsymbol{b}}_j)(\boldsymbol{e}_i^\mathsf{T}-\boldsymbol{e}_j^\mathsf{T}) \tag{10}$$

where $\tilde{\boldsymbol{b}}_i$ denotes the i-th column of $\tilde{\boldsymbol{B}}$, and $\boldsymbol{e}_i$ is the i-th column of the identity matrix $\boldsymbol{I}\in\mathbb{R}^{N\times N}$. Inserting (10) into the cost function (9), we obtain:

$$\begin{aligned}
J_2(\theta_{i,j}) &= \sum_{k=1}^{K}\left\|\operatorname{off}\left(\tilde{\boldsymbol{C}}^{(k,\text{new})}\right)\right\|_F^2\\
&= \sum_{k=1}^{K}\Big\|\operatorname{off}\big([(\boldsymbol{Q}^{(i,j)}(\theta_{i,j}))^{\boxdot 2}]^\mathsf{T}\tilde{\boldsymbol{C}}^{(k)}(\boldsymbol{Q}^{(i,j)}(\theta_{i,j}))^{\boxdot 2}\\
&\quad+\sin(2\theta)[(\boldsymbol{Q}^{(i,j)}(\theta_{i,j}))^{\boxdot 2}]^\mathsf{T}\tilde{\boldsymbol{c}}^{(k,1)}(\boldsymbol{e}_i^\mathsf{T}-\boldsymbol{e}_j^\mathsf{T})\\
&\quad+\sin(2\theta)(\boldsymbol{e}_i-\boldsymbol{e}_j)\tilde{\boldsymbol{c}}^{(k,2)}(\boldsymbol{Q}^{(i,j)}(\theta_{i,j}))^{\boxdot 2}\\
&\quad+\sin^2(2\theta)\tilde{c}^{(k,3)}(\boldsymbol{e}_i-\boldsymbol{e}_j)(\boldsymbol{e}_i^\mathsf{T}-\boldsymbol{e}_j^\mathsf{T})\big)\Big\|_F^2
\end{aligned} \tag{11}$$

where $\tilde{\boldsymbol{C}}^{(k)}=\tilde{\boldsymbol{A}}^\mathsf{T}\boldsymbol{C}^{(k,-1)}\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{c}}^{(k,1)}=\tilde{\boldsymbol{A}}^\mathsf{T}\boldsymbol{C}^{(k,-1)}(\tilde{\boldsymbol{b}}_i\boxdot\tilde{\boldsymbol{b}}_j)$, $\tilde{\boldsymbol{c}}^{(k,2)}=(\tilde{\boldsymbol{c}}^{(k,1)})^\mathsf{T}$ and $\tilde{c}^{(k,3)}=(\tilde{\boldsymbol{b}}_i\boxdot\tilde{\boldsymbol{b}}_j)^\mathsf{T}\boldsymbol{C}^{(k,-1)}(\tilde{\boldsymbol{b}}_i\boxdot\tilde{\boldsymbol{b}}_j)$ are a matrix, a column vector, a row vector and a scalar of constant values, respectively. (11) shows that just the i-th and j-th columns and rows of $\tilde{\boldsymbol{C}}^{(k,\text{new})}$ involve the parameter $\theta_{i,j}$. It is noteworthy that the (i,j)-th and (j,i)-th elements are twice affected by the transformation. Inspired by [12], we propose to minimize the sum of the squares of the (i,j)-th entries of the K symmetric matrices $\tilde{\boldsymbol{C}}^{(k,\text{new})}$, instead of minimizing all the off-diagonal entries. This simplified minimization criterion is denoted by $\tilde{J}_2(\theta_{i,j})$. The (i,j)-th element of $\tilde{\boldsymbol{C}}^{(k,\text{new})}$ can be expressed as a function of $\theta_{i,j}$ as follows:

$$\begin{aligned}
\tilde{C}_{i,j}^{(k,\text{new})} &= -\sin^2(2\theta_{i,j})\tilde{c}^{(k,3)}\\
&\quad+\sin^2(\theta_{i,j})(\tilde{C}_{i,i}^{(k)}\cos^2(\theta_{i,j})+\tilde{C}_{j,i}^{(k)}\sin^2(\theta_{i,j}))\\
&\quad+\cos^2(\theta_{i,j})(\tilde{C}_{i,j}^{(k)}\cos^2(\theta_{i,j})+\tilde{C}_{j,j}^{(k)}\sin^2(\theta_{i,j}))\\
&\quad+\sin(2\theta_{i,j})(\tilde{c}_i^{(k,1)}\cos^2(\theta_{i,j})+\tilde{c}_j^{(k,1)}\sin^2(\theta_{i,j}))\\
&\quad-\sin(2\theta_{i,j})(\tilde{c}_j^{(k,2)}\cos^2(\theta_{i,j})+\tilde{c}_i^{(k,2)}\sin^2(\theta_{i,j}))
\end{aligned} \tag{12}$$

where $\tilde{C}_{i,j}^{(k)}$ is the (i,j)-th element of $\tilde{\boldsymbol{C}}^{(k)}$ and $\tilde{c}_i^{(k,q)}$ is the i-th element of vector $\tilde{\boldsymbol{c}}^{(k,q)}$ with $q\in\{1,2\}$. By using the Weierstrass change of variable: $t_{i,j}=\tan(\theta_{i,j})$, the expression of (12) can be rewritten as follows:

$$\tilde{C}_{i,j}^{(k,\text{new})}=\frac{f_4^{(k)}t_{i,j}^4+f_3^{(k)}t_{i,j}^3+f_2^{(k)}t_{i,j}^2+f_1^{(k)}t_{i,j}+f_0^{(k)}}{(1+t_{i,j}^2)^2} \tag{13}$$

where $f_4^{(k)}=\tilde{C}_{j,i}^{(k)}$, $f_3^{(k)}=-2\tilde{c}_i^{(k,1)}$, $f_2^{(k)}=\tilde{C}_{i,i}^{(k)}+\tilde{C}_{j,j}^{(k)}+2\tilde{c}_j^{(k,2)}-4\tilde{c}^{(k,3)}$, $f_1^{(k)}=2\tilde{c}_i^{(k,2)}-\tilde{c}_j^{(k,1)}$ and $f_0^{(k)}=\tilde{C}_{j,j}^{(k)}$. Equation (13) shows that the sum of the squares of $\tilde{C}_{i,j}^{(k,\text{new})}$, is a rational function in $t_{i,j}$, namely $\tilde{J}_2(t_{i,j})$, where the degrees of the numerator and the denominator are 8 and 8, respectively. The global minimum $t_{i,j}$ can be obtained by computing the roots of its derivative and selecting the one yielding the smallest value of $\tilde{J}_2(t_{i,j})$. Once $t_{i,j}$ is obtained, $\theta_{i,j}$ can be computed by $\theta_{i,j}=\arctan(t_{i,j})$. Then $\tilde{\boldsymbol{B}}$ is updated by (8) and $\tilde{\boldsymbol{A}}$ is updated by computing $(\tilde{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$.

B. Minimization with respect to $\boldsymbol{R}^{(i,j)}(r_{i,j})$

Let $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ continue to denote the current estimate of $\boldsymbol{A}$ and $\boldsymbol{B}$ before estimating $\boldsymbol{R}^{(i,j)}(r_{i,j})$, respectively. The update of $\tilde{\boldsymbol{B}}$, denoted by $\tilde{\boldsymbol{B}}^{(\text{new})}$, is defined as follows:

$$\tilde{\boldsymbol{B}}^{(\text{new})} = \tilde{\boldsymbol{B}}\boldsymbol{R}^{(i,j)}(r_{i,j}) \tag{14}$$

By replacing matrix $\tilde{\boldsymbol{B}}$ by $\tilde{\boldsymbol{B}}^{(\text{new})}$ into criterion (6), the criterion $J_2(r_{i,j})$ can be expressed as follows:

$$J_2(r_{i,j}) = \sum_{k=1}^{K} \left\| \text{off}\left\{ [(\tilde{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}]^{\mathsf{T}} \boldsymbol{C}^{(k,-1)} [(\tilde{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}] \right\} \right\|_F^2 \tag{15}$$

The Hadamard square of $\tilde{\boldsymbol{B}}^{(\text{new})}$ in (15) can be expressed as a function of $r_{i,j}$ as follows:

$$(\tilde{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} = \tilde{\boldsymbol{B}}^{\boxdot 2}\boldsymbol{R}^{(i,j)}(r_{i,j}^2) + 2\, r_{i,j}(\tilde{\boldsymbol{b}}_i \boxdot \tilde{\boldsymbol{b}}_j)\boldsymbol{e}_j^{\mathsf{T}} \tag{16}$$

where $\tilde{\boldsymbol{b}}_i$ denotes the i-th column of $\tilde{\boldsymbol{B}}$, and $\boldsymbol{e}_j$ is the j-th column of the identity matrix $\boldsymbol{I} \in \mathbb{R}^{N\times N}$. Inserting (16) into the cost function (15), we have:

$$\begin{aligned} J_2(r_{i,j}) &= \sum_{k=1}^{K} \|\, \text{off}\left(\tilde{\boldsymbol{C}}^{(k,\text{new})}\right)\|_F^2 \\ &= \sum_{k=1}^{K} \Big\| \text{off}\,\Big(\boldsymbol{R}^{(i,j)}(r_{i,j}^2)^{\mathsf{T}} \tilde{\boldsymbol{C}}^{(k)} \boldsymbol{R}^{(i,j)}(r_{i,j}^2) + r_{i,j}^2 \tilde{c}^{(k,3)} \boldsymbol{e}_j \boldsymbol{e}_j^{\mathsf{T}} \\ &\quad + r_{i,j} \boldsymbol{R}^{(i,j)}(r_{i,j}^2)^{\mathsf{T}} \tilde{\boldsymbol{c}}^{(k,1)} \boldsymbol{e}_j^{\mathsf{T}} + r_{i,j} \boldsymbol{e}_j \tilde{\boldsymbol{c}}^{(k,2)} \boldsymbol{R}^{(i,j)}(r_{i,j}^2)\Big)\Big\|_F^2 \end{aligned} \tag{17}$$

where $\tilde{\boldsymbol{C}}^{(k)} = \tilde{\boldsymbol{A}}^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{c}}^{(k,1)} = 2\tilde{\boldsymbol{A}}^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}(\tilde{\boldsymbol{b}}_i \boxdot \tilde{\boldsymbol{b}}_j)$, $\tilde{\boldsymbol{c}}^{(k,2)} = (\tilde{\boldsymbol{c}}^{(k,1)})^{\mathsf{T}}$ and $\tilde{c}^{(k,3)} = 4\,(\tilde{\boldsymbol{b}}_i \boxdot \tilde{\boldsymbol{b}}_j)^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}(\tilde{\boldsymbol{b}}_i \boxdot \tilde{\boldsymbol{b}}_j)$ are a matrix, a column vector, a row vector and a scalar of constant values, respectively. (17) shows that just the j-th column and row of $\tilde{\boldsymbol{C}}^{(k,\text{new})}$ involve the parameter $r_{i,j}$. Therefore, the minimization of the cost function (17) is equivalent to minimizing the sum of the squares of the j-th columns of all the symmetric matrices $\tilde{\boldsymbol{C}}^{(k,\text{new})}$ except their (j,j)-th elements. These elements can be expressed by a polynomial function of degree 2 in $r_{i,j}$ as follows, for every n value different of j:

$$\tilde{C}_{n,j}^{(k,\text{new})} = \tilde{C}_{n,i}^{(k)} r_{i,j}^2 + \tilde{c}_n^{(k,1)} r_{i,j} + \tilde{C}_{n,j}^{(k)} \tag{18}$$

where $\tilde{C}_{n,i}^{(k)}$ is the (n,i)-th component of $\tilde{\boldsymbol{C}}^{(k)}$, and $\tilde{c}_n^{(k,1)}$ is the n-th element of $\tilde{\boldsymbol{c}}^{(k,1)}$. Then the cost function (17), which is the total sum of squares of (18), is a polynomial function of degree 4 in $r_{i,j}$. The global minimum $r_{i,j}$ is one of the roots of its derivative, which yields the smallest value of (17). Once the optimal $r_{i,j}$ is computed, $\tilde{\boldsymbol{B}}$ is updated by (14) and $\tilde{\boldsymbol{A}}$ is updated by computing $(\tilde{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$.

The processing of all the $N(N-1)$ parameters $\theta_{i,j}$ and $r_{i,j}$, is called a QR sweep. The proposed $\text{JD}_{\text{QR}}^{+}$ algorithm is comprised of several QR sweeps in order to guarantee the convergence. In ICA, when a non-square matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times P}$ with $N > P$ is encountered, we can compress it by a matrix $\boldsymbol{W} \in \mathbb{R}_{+}^{N\times P}$ such that the resulting matrix $\bar{\boldsymbol{A}} = \boldsymbol{W}^{\mathsf{T}}\boldsymbol{A}$ is a nonnegative square matrix [15]. It is noteworthy that the proposed algorithm is different from the two published nonnegative JDC methods, which are based on the LU matrix factorization [10], [11]. We use QR factorization in this paper. The method in [10] estimates $\boldsymbol{B}$ and $\boldsymbol{D}^{(k)}$ alternately, and its performance is sensitive to the initialization. The algorithm in [11] needs to compute the inverse of $\boldsymbol{A}$ in all the $N(N-1)$ Jacobi-like iterations, leading to a high numerical complexity.

III. Simulation results

In this section, the proposed $\text{JD}_{\text{QR}}^{+}$ algorithm is compared with several existing JDC methods and BSS algorithms. The performance is measured in terms of the error between the true matrix $\boldsymbol{A}$ and its estimate $\tilde{\boldsymbol{A}}$, as well as the source $\boldsymbol{s}$ and its estimate $\tilde{\boldsymbol{s}}$ when a BSS context is considered. The following scale-invariant and permutation-blind distance is chosen as the preferred measure:

$$\alpha(\boldsymbol{A}, \tilde{\boldsymbol{A}}) = ({}^{1}/{}_{N}) \sum_{n=1}^{N} \min_{(n,n')\in I_n^2} d(\boldsymbol{a}_n, \tilde{\boldsymbol{a}}_{n'}) \tag{19}$$

where $\boldsymbol{a}_n$ and $\tilde{\boldsymbol{a}}_{n'}$ are the n-th column of $\boldsymbol{A}$ and the n'-th column of $\tilde{\boldsymbol{A}}$, respectively. I_n^2 is defined recursively by $I_1^2 = \{1,\cdots,N\} \times \{1,\cdots,N\}$, and $I_{n+1}^2 = I_n^2 - J_n^2$, where $J_n^2 = \text{argmin}_{(n,n')\in I_n^2} d(\boldsymbol{a}_n, \tilde{\boldsymbol{a}}_{n'})$. In addition, $d(\boldsymbol{a}_n, \tilde{\boldsymbol{a}}_{n'})$ is defined as the pseudo-distance between two vectors [4]:

$$d(\boldsymbol{a}_n, \tilde{\boldsymbol{a}}_{n'}) = 1 - \|\boldsymbol{a}_n^{\mathsf{T}}\,\tilde{\boldsymbol{a}}_{n'}\|^2 / (\|\boldsymbol{a}_n\|^2 \|\tilde{\boldsymbol{a}}_{n'}\|^2) \tag{20}$$

The smaller the value of (19) is, the better estimation of $\boldsymbol{A}$ is achieved.

A. Simulated semi-nonnegative semi-symmetric arrays

In this part, $\text{JD}_{\text{QR}}^{+}$ is compared with two classic JDC methods, namely ACDC [5] and FFDIAG [6], and one nonnegative JDC method $\text{ACDC}_{\text{LU}}^{+}$ [10] with simulated semi-nonnegative semi-symmetric 3-way arrays $\boldsymbol{\mathcal{C}}$. $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{3\times 3\times 5}$ is generated randomly according to equation (2). The loading matrices $\boldsymbol{A}$ and $\boldsymbol{D}$ are randomly drawn from a uniform distribution between 0 and 1. The pure array $\boldsymbol{\mathcal{C}}$ is perturbed by a semi-symmetric residual noise array $\boldsymbol{\mathcal{V}}$. The loading matrices of $\boldsymbol{\mathcal{V}}$ obey the zero-mean unit-variance Gaussian distribution. The resulting noisy 3-way array can be written by $\boldsymbol{\mathcal{C}}_N = \boldsymbol{\mathcal{C}}/\|\boldsymbol{\mathcal{C}}\|_F + \sigma_N \boldsymbol{\mathcal{V}}/\|\boldsymbol{\mathcal{V}}\|_F$, where σ_N is a scalar controlling the noise level. Then the Signal-to-Noise Ratio (SNR) is defined by $\text{SNR} = -20\log_{10}(\sigma_N)$. All the algorithms stop either when the relative error of the corresponding criterion between two successive sweeps is less than 10^{-5} or when the number of sweeps exceeds 200. We repeat the experiment with SNR ranging from -10 dB to 30 dB with 500 Monte Carlo trials. Figure 1 shows the average curves of $\alpha(\boldsymbol{A}, \tilde{\boldsymbol{A}})$ of all the three algorithms as a function of SNR. It shows that ACDC performs better than FFDIAG under higher SNR levels. The nonnegativity constraint obviously helps $\text{ACDC}_{\text{LU}}^{+}$ and $\text{JD}_{\text{QR}}^{+}$ to outperform the classic ones. The proposed $\text{JD}_{\text{QR}}^{+}$ algorithm maintains the best estimation accuracy, especially for the lower SNR levels.

B. BSS application on MRS data

In this section, the BSS performance of $\text{JD}_{\text{QR}}^{+}$ is compared with an effective ICA method CoM_2 [16] and a Nonnegative Matrix Factorization (NMF) method based on alternating Non-Negativity Least Squares (NNLS) [17], through an experiment carried out on simulated MRS data. Two metabolites, namely the Choline and Myo-inositol, serve as source signals $s_1(f)$ and $s_2(f)$. 32 observations are generated according to the noisy linear mixing model $\boldsymbol{x}(f) = \boldsymbol{A}\boldsymbol{s}(f) + \boldsymbol{\nu}(f)$, where $\boldsymbol{\nu}(f)$ is an additive white Gaussian noise. $\boldsymbol{A} \in \mathbb{R}_{+}^{32\times 2}$ is similarly generated as in the previous section. For an ICA method based

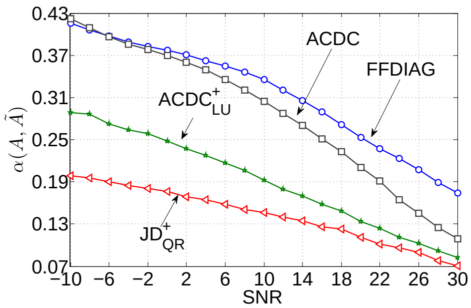


Fig. 1. Average error $\alpha(\boldsymbol{A}, \tilde{\boldsymbol{A}})$ evolution of ACDC, FFDIAG, $\mathrm{ACDC}^{+}_{\mathrm{LU}}$ and $\mathrm{JD}^{+}_{\mathrm{QR}}$ as a function of SNR on simulated arrays.

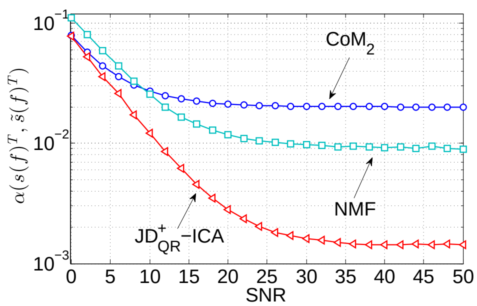
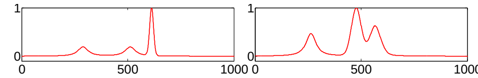
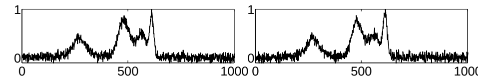


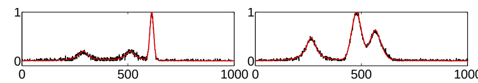
Fig. 2. Average error $\alpha(\{\boldsymbol{s}(f)\}^{\mathsf{T}}, \{\tilde{\boldsymbol{s}}(f)\}^{\mathsf{T}})$ evolution of CoM_2, NMF and $\mathrm{JD}^{+}_{\mathrm{QR}}$-ICA as a function of SNR for BSS of 2 simulated MRS metabolites.



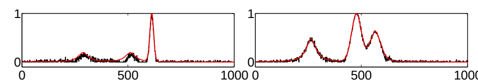
(a) Choline and Myo-inositol source metabolites



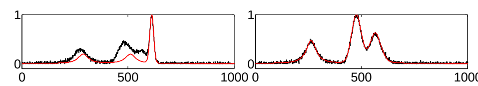
(b) Two of the observations



(c) Estimates by $\mathrm{JD}^{+}_{\mathrm{QR}}$-ICA



(d) Estimates by CoM_2



(e) Estimates by NMF

Fig. 3. MRS source metabolites, observations and estimated metabolites by $\mathrm{JD}^{+}_{\mathrm{QR}}$-ICA, CoM_2 and NMF with SNR $= 10$ dB for one typical realization. The overlapping red lines in figures (c), (d) and (e) indicate the correct sources.

on $\mathrm{JD}^{+}_{\mathrm{QR}}$, namely $\mathrm{JD}^{+}_{\mathrm{QR}}$-ICA, $\{\boldsymbol{x}(f)\}$ is compressed by means of a matrix $\boldsymbol{W} \in \mathbb{R}^{32\times 2}_{+}$ computed using the method proposed in [15], such that the number of observations is reduced to 2. The 3-way array $\boldsymbol{\mathcal{C}}$ is built by stacking four 4-th order cumulant matrix slices. We repeat the experiment with SNR ranging from 0 dB to 50 dB with 200 Monte Carlo trials. The average curves of the estimating error $\alpha(\{\boldsymbol{s}(f)\}^{\mathsf{T}}, \{\tilde{\boldsymbol{s}}(f)\}^{\mathsf{T}})$ of all the three methods as a function of SNR are shown in figure 2. It shows that the proposed $\mathrm{JD}^{+}_{\mathrm{QR}}$-ICA algorithm maintains competitive advantages when SNR ≥ 5 dB. Figure 3 shows the separation results of all the methods with a SNR of 10 dB for one typical realization. Regarding CoM_2 and NMF, there are some obvious disturbances presented in the estimated metabolites. As far as $\mathrm{JD}^{+}_{\mathrm{QR}}$-ICA is concerned, the estimated source metabolites are quasi-perfect.

IV. Conclusion

In this paper, we have addressed the problem of the CP decomposition of semi-nonnegative semi-symmetric 3-way arrays. We proposed a method, called $\mathrm{JD}^{+}_{\mathrm{QR}}$, based on the QR factorization of the Hadamard square root of the nonnegative loading matrix. A numerical experiment on simulated arrays highlights its advantage. A BSS application on MRS signals also demonstrates the interest of the proposed method.

References

[1] X. Luciani and L. Albera, "Canonical polyadic decomposition based on joint eigenvalue decomposition," *Chemometr. Intell. Lab.*, vol. 132, pp. 152–167, 2014.

[2] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis: Applications in the Chemical Sciences*. West Sussex, England: Wiley, 2004.

[3] P. Comon, X. Luciani, and A. L. F. de Almeida, "Tensor decompositions, alternating least squares and other tales," *J. Chemometr.*, vol. 23, pp. 393–405, 2009.

[4] L. Albera, A. Ferréol, P. Comon, and P. Chevalier, "Blind identification of overcomplete mixtures of sources (BIOME)," *Linear Algebra Appl.*, vol. 391, pp. 3–30, 2004.

[5] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Trans. Signal Process.*, vol. 50, no. 7, pp. 1545–1553, 2002.

[6] A. Ziehe, P. Laskov, G. Nolte, and K.-R. Muller, "A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation," *J. Mach. Learning Res.*, vol. 5, pp. 777–800, 2004.

[7] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. West Sussex, United Kingdom: WILEY, 2009.

[8] A. Shashua, R. Zass, and T. Hazan, "Multi-way clustering using super-symmetric non-negative tensor factorization," in *Computer Vision-ECCV*, 2006, pp. 595–608.

[9] P. Tichavský and Z. Koldovský, "Weight adjusted tensor method for blind separation of underdetermined mixtures of nonstationary sources," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1037–1047, 2011.

[10] L. Wang, L. Albera, A. Kachenoura, H. Shu, and L. Senhadji, "Nonnegative joint diagonalization by congruence based on LU matrix factorization," *IEEE Signal Process. Lett.*, vol. 20, no. 8, pp. 807–810, 2013.

[11] L. Wang, L. Albera, H. Shu, and L. Senhadji, "A new Jacobi-like nonnegative joint diagonalization by congruence," in *Proceedings of the XXI European Signal Processing Conference (EUSIPCO'13)*, 2013, paper id: 1569744591.

[12] A. Souloumiac, "Nonorthogonal joint diagonalization by combining Givens and hyperbolic rotations," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2222–2231, 2009.

[13] M. Chu, F. Diele, R. Plemmons, and S. Ragni, "Optimality computation and interpretation of non negative matrix factorizations," Wake Forest University, Tech. Rep., 2004.

[14] J.-P. Royer, P. Comon, and N. Thirion-Moreau, "Nonnegative 3-way tensor factorization via conjugate gradient with globally optimal stepsize," in *ICASSP'11*, 2011, pp. 4040–4043.

[15] L. Wang, A. Kachenoura, L. Albera, H. Shu, and L. Senhadji, "Nonnegative compression for semi-nonnegative independent component analysis," submitted to IEEE SAM 2014 Workshop.

[16] P. Comon, "Independent component analysis, a new concept?" *Signal Process.*, vol. 36, no. 3, pp. 287–314, 1994.

[17] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 2, pp. 713–730, 2008.